

Abusing Publish and Subscribe

Idea

Qt currently has no cross-platform API for inter process communication. However, there is now a mechanism called "Publish and Subscribe" that enables one application to publish so called "values" that another application can subscribe to. The application that publishes values notifies each subscriber about changes. The application that subscribed to a value can then update the display.

For example, a media player could publish the currently played track and an applet on the homescreen could subscribe to that value. This can be implemented in a cross platform manner now. A sample implementation for the iTunes mediaplayer on Mac OS is linked at the bottom of this page.

In the end, the publisher can invoke a method in the subscriber. Everytime the publisher updates the value, a method is called in the subscriber. It would be nice if this mechanism could be used to control the media player from the homescreen. A typical usecase for such a control would be to pause and resume media playback.

Communication using Publish and Subscribe only works in one direction, so the homescreen applet has to publish something that the mediaplayer can subscribe. Each time the user toggles between play and pause, a published value of the homescreen applet has to change in order to invoke a method in the media player. The published value could be simply a boolean that flips back and forth or it could be also the number of times the button has been pressed since the applet was started.

By using this technique, two-way communication can be implemented using Publish and Subscribe.

Example

The attached example shows how to control a mediaplayer. The publisher shows only a playbutton and published how often the button was clicked. The subscriber talks to the mediaplayer and shows the currently played title with a nice animated GUI with reflections. The example controls the iTunes media player on Mac OS. The example might work on other platforms as well but does not control a real mediaplayer. On Mac OS, it was not possible to establish a two way communication. I guess there is an implementation detail that prevents an application from both publishing and subscribing values.

Download

[File:AbusingPublishAndSubscribe.zip](#)

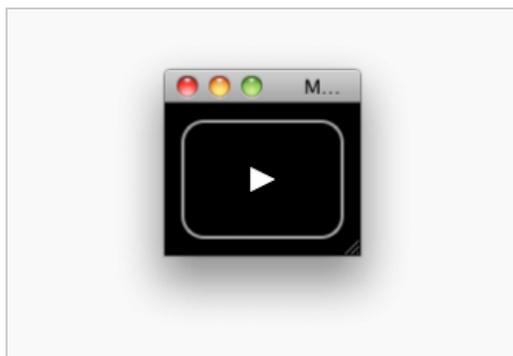
Compiling

Both the publisher and subscriber are simply compiled using `qmake && make`.

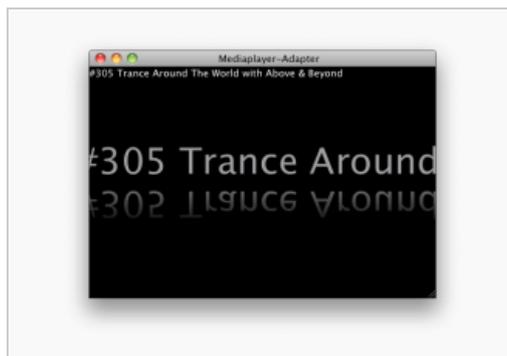
Running the example

First start the publisher, then the subscriber.

Screenshots



Screenshot of publisher



Screenshot of subscriber

